



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

MOBILNÍ APLIKACE PRO PRÁCI S UNIVERZITNÍM ELEARNINGOVÝM PORTÁLEM NA PLATFORMĚ WINDOWS PHONE 8

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Martin Pomezný**

Vedoucí práce: Ing. Igor Kopetschke





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

MOBILE APPLICATION FOR E-LEARNING PORTAL ON WINDOWS PHONE 8

Diploma thesis

Study programme: N2612 – Electrotechnology and informatics

Study branch: 1802T007 – Information technology

Author: **Bc. Martin Pomezný**

Supervisor: Ing. Igor Kopetschke



Tento list nahradte
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Datum:

Podpis:

Abstrakt

Cílem této práce bylo nastudovat si principy fungování e-learningového systému Moodle a navrhnout vhodnou a především bezpečnou formu komunikace s mobilní aplikací na platformě Windows Phone 8. Tato aplikace byla také naprogramována a umožňuje studentům využívat služeb portálu na telefonu v přehledném a intuitivním prostředí. Velký důraz byl kladen na bezpečnost komunikace, protože na serveru jsou uložena i citlivá data o uživateli. Jakou součástí práce také vznikla komponenta v jazyce C# zpřístupňující webové služby. Ta jde velmi snadno rozšířit, nebo použít v jiném projektu.

Aplikace komunikuje s portálem přes zabezpečený protokol HTTPS přes síť internet. Spojení je realizováno webovou službou, která využívá protokolu REST. V rámci práce byla studentům zpřístupněna i možnost přehrávat přednášky ze systému Mediasite na mobilním telefonu. Při tom bylo zdokumentováno zabezpečení streamů ze zmiňovaného systému.

Klíčová slova: Moodle, C#, Windows Phone, webové služby, REST

Abstract

Goal of this thesis was to study functional principles of e-learning system Moodle and to propose suitable and especially safe form of communication with the mobile application on Windows Phone 8 platform. This application was also programmed and it allows students to use the features of the portal on their cell phone in a well-arranged and intuitive interface. Big emphasis was put on the safety of communication, because the user data stored on the server are sensitive nature. A component in C# language was also created as a part of this thesis study, which implements web services functions. This component might be extended very easily or can be used in another project.

The application communicates with the portal via safe protocol HTTPS through internet. The connection is built on web services, which uses REST protocol. The possibility of streaming the lectures from Mediasite system on the cell phone was made accessible to students thanks to this thesis study. Doing that, the protection of streaming from above mentioned system was also documented.

Keywords: Moodle, C#, Windows Phone, web services, REST

Poděkování

Děkuji především vedoucímu práce Ing. Igoru Kopetschkemu za metodické vedení a cenné rady během tvorby této práce.

Děkuji své rodině a přátelům za to, že mi byli oporou a podporovali mě po celou dobu mého studia.

Obsah

Seznam zkratek	9
Úvod	10
1 E-learning na TUL	13
1.1 Mediasite	15
1.2 Bezpečnost	16
2 Webové služby a spojení klient-server	17
2.1 Volba formátu	18
2.1.1 SOAP	18
2.1.2 XML-RPC	19
2.1.3 Action message format	19
2.1.4 REST	19
2.2 Server	20
2.2.1 Konfigurace	20
2.2.2 Vývoj	22
2.3 Klient	23
2.3.1 Autentizace a token	24
2.3.2 Framework Json.NET	26
2.3.3 Funkce	26
3 Aplikace	30
3.1 Izolované úložiště	31
3.2 Zápis předmětů ze STAGu	32
3.3 Komunikace se serverem	33
3.4 Přehrávání z Mediasite	36
3.4.1 Zabezpečení	37
3.4.2 Přehrávání v aplikaci	39
3.5 Navigace	41
3.6 Uspořádání	42
3.7 Distribuce	48
Závěr	49
Výsledky	49
Rozšíření a úpravy	51
Podobné projekty	52

Seznam zkratek

HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
REST	Representational State Transfer
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
TUL	Technická univerzita v Liberci
UI	User Interface – Uživatelské rozhraní
URI	Uniform Resource Identifier
WP8	Windows Phone 8
XML	Extensible Markup Language

Úvod

Během posledních několika let zasahují do světa informačních technologií nová zařízení, vyznačující se malým displayem s dotykovou vrstvou se kterou uživatelé pracují prsty. Řeč je o tzv. mobilních zařízeních. Těm vévodí „chytré telefony“, tedy mobilní telefony s operačním systémem a možností instalovat nejrůznější aplikace rozšiřující funkce zařízení. Těchto telefonů se v roce 2011 poprvé prodalo více než osobních počítačů [2] což jen potvrzuje jejich popularitu.

Tato zvyšující se obliba mění podobu webu a je v současnosti nutné s tímto obrovským segmentem trhu počítat a přizpůsobovat web jako takový potřebám mobilních telefonů. Jejich specifika kladou na webové stránky jiné požadavky, protože display je oproti počítačovému monitoru malý a aby byl obsah čitelný, musí jej uživatel zvětšit. To ale způsobí, že se na display nevejde celá šíře stránky a uživatel tak má zkomplikovanou navigaci po stránce a pro některé to je značně matoucí, protože lidé jsou zvyklí na čtení a prohlížení obsahu směrem odshora dolů. Taktéž ovládací prvky nejsou uzpůsobeny pro velkou plochu konečku prstu, ale pro přesný pohyb počítačovou myší.

K řešení uvedených problémů je možno přistoupit několika směry. Prvním možným je celý web vytvořit znovu s optimalizací pro mobilní zařízení. Tato varianta byla populární především dříve a vyznačovala se přesměrováním mobilních zařízení na adresu která začínala „m.“. Tato metoda je ale značně náročná na údržbu a v případě změn je nutné ji provádět na více místech. Alternativně se vyvinula metoda druhá, využívající faktu, že webové stránky by měly mít oddělený obsah

od vzhledu. To umožní pro různá zařízení zobrazovat různou formu obsahu, aniž by vznikla potřeba upravovat funkční část stránek, protože změny je dosahováno úpravami kaskádových stylů, případně ve spojení s JavaScriptem. S tímto postupem úzce souvisí pojem *responzivní web design*. Ten definuje možnost dynamicky škálovat obsah pomocí jednoho stylu a přeskládat jej, aby bylo umožněno čtení směrem odshora dolů a obsah byl stále čitelný a použitelný. Další možností je vytvoření aplikace pro konkrétní potřeby webu. To pochopitelně není vhodné řešení pro všechny weby, ale spíše pro komplexní systémy nebo weby poskytující služby, kde aplikace skutečně usnadní uživateli používání a přinese mu jistý komfort. Taková aplikace navíc může využít přirozených možností konkrétního systému.

Portál Moodle nevyužívá responzivního web designu a předělání tak velkého a komplexního projektu by bylo velmi náročné. Ačkoli se o to tým vyvíjející Moodle již delší dobu pokouší o mobilní vzhled webu, stále je z velké části nedokončený a i proto jej administrátoři nezapínají. Uživatelé si tedy musí vystačit s jeho vzhledem designovaným pro osobní počítače i na mobilních telefonech. Částečně i proto vzniklo zadání této práce, která si klade za cíl umožnit uživatelům platformy Windows Phone 8 využívat pohodlně služeb e-learningu.

Systém Windows Phone navazuje na dlouholetou tradici operačních systémů pro mobilní telefony společnosti Microsoft. Po ukončení vývoje Windows Mobile roku 2010 přišla společnost se systémem Windows Phone 7, který je kompletně přepracován a od původního se liší celkovým pojetím koncepce „chytrého telefonu“. Zatímco původní systém se velmi podobal Windows z osobních počítačů a připomínal spíše malý počítač, nový jde opačnou cestou a telefon prezentuje jako funkční řešení, které sice nemá tolik možností nastavení a úprav, ale cílí na rychlost, přehlednost a tím především na laické uživatele. Windows Phone 8 je již druhou iterací systému a nahrazuje na trhu Windows Phone 7.

Ačkoli v prvních třech letech prodeje telefonů s Windows Phone stagnovaly, s vydáním WP8 se odrazily ode dna a společnost Gartner dokonce odhaduje, že se

v roce 2016 v počtu prodaných zařízení dostane před operační systém iOS společnosti Apple [3].

Před samotnou tvorbou aplikace bylo nutné se seznámit s e-learningovým portálem Moodle, jeho funkcemi a odlišnostmi, které jsou specifické pro TUL a také vybrat vhodnou implementaci webových služeb. Vzhledem k velikosti celého portálu a povaze uložených dat byl kladen důraz i na bezpečnost a co nejmenší možný počet úprav a zásahů do systému.

Po seznámení a konfiguraci testovacího prostředí vznikla softwarová komponenta v jazyce C#. Ta implementuje funkce webové služby a dá se po drobných úpravách použít i při tvorbě jiné aplikace. Po zevrubném otestování komponenty byla zahájena práce na mobilní aplikaci, která ji využívá pro komunikaci se systémem Moodle.

Uživatelé se do aplikace přihlašují jménem a heslem, které si můžou i nechat zapamatovat, aby je nemuseli stále znova vyplňovat. Přihlašovací údaje jsou v případě TUL shodné s přístupy k síti LIANE. Po přihlášení mohou studenti procházet své kurzy a prohlížet materiály nahrané vyučujícím. V aplikaci také funguje přehrávání záznamů přednášek z portálu Mediasite umístěném na adrese <http://prednasky.tul.cz>.

1. E-learning na TUL

E-learning byl a částečně stále je na TUL roztržštěn do mnoha drobných portálů pro studenty spravovaných jednotlivými fakultami nebo vyučujícími daných předmětů. Z části jsou využívány systémy jako je Moodle, někdy se dokonce jedná o emailovou schránku, do které mají studenti přístup. Tato roztržštěná řešení s sebou přináší řadu nevýhod a komplikací, jako je například nutnost si pamatovat větší množství adres a přístupových údajů. Taktéž sledovat aktuální dění (například zrušené cvičení nebo přednáška) vyžaduje nemalou iniciativu studenta.

Vzhledem k výše uvedenému je pochopitelná snaha zavést jednotný e-learningový portál, který usnadní používání studentům i pedagogům, zlepší dostupnost materiálů a centralizuje správu alespoň na úrovni jednotlivých fakult. Jako e-learningový portál byl zvolen open-source software Moodle vydaný pod licencí GNU GPL. To zaručuje, že jeho použití je zdarma, jeho zdrojový kód je otevřený a lze snadno upravit pro konkrétní potřeby školy. Mezi základní vlastnosti a výhody patří:

- Jak udává oficiální statistika [4] celkový počet registrovaných instalací dosahuje 68 000 ve více než 200 zemích světa. Díky tomu má velkou komunitu a potenciál dalšího vývoje a opravy případných objevených chyb.
- Díky množství pluginů je rozšiřitelný o další funkce a datové formáty, které přijímá nebo produkuje.

- Je připraven pro integraci do stávajících systémů a infrastruktury organizace. Autentizace může být prováděna nad stávajícími databázemi nebo proti již běžícím službám jakou může být například centrální LDAP.
- Kurzy mohou být vytvářeny přímo v Moodle nebo mohou být importovány v množství různých formátů, mezi nimiž nechybí ani standardizovaný referenční model SCORM.
- Jednotnost uspořádání kurzů umožňuje přehledný přístup ke všem studijním materiálům a úkolům uživatele v roli studenta i pedagoga.
- Prostředí je uživatelsky přívětivé a přehledné, za zmínku stojí možnost vytvářet i komplexní kurzy s anketami, testy a úkoly jen se základními znalostmi práce s počítačem a webovou stránkou.

Aby nedošlo k narušení chodu stávajícího portálu v důsledku testování a nutnosti vlastnit správcovské oprávnění, bylo nutné nainstalovat testovací instanci Moodle na jiném počítači a přiblížit se jejím nastavením oficiálnímu portálu školy. Serverová část Moodle je napsaná v PHP a pro svůj chod tedy potřebuje interpret jazyka PHP, webový server Apache a jako datové úložiště open-source relační databázový systém MySQL (nebo MariaDB, což je komunitou spravovaná větev MySQL). Pro operační systém Windows tyto tři programy vhodně kombinuje balík WAMP (Windows, Apache, MySQL, PHP). Po jeho nainstalování již bylo možné stáhnout instalaci Moodle z oficiálního zdroje <http://download.moodle.org/>. Pro potřeby vývoje byla použita aktuální stabilní verze, kterou byla toho času verze 2.6 (Build: 20131118). Klientská strana, kterou je internetový prohlížeč, není třeba nijak upravovat, protože je použita zcela standardní technologie HTML a JavaScript.

1.1 Mediasite

Mediasite je soubor technologií společnosti Sonic Foundry Inc., které umožňují interaktivně vytvářet audiovizuální díla záznamem dění v místnosti a posléze je prezentovat v uživatelsky přívětivé formě. Na TUL se využívá především pro záznam přednášek, ale i zasedání či besed. Systém sestává z několika částí, kde první je záznamový hardware v učebně. Ten tvoří dotykové rozhraní, jeden či více projektorů, mikrofón pro přednášejícího, reproduktory, kamera a v neposlední řadě magnetická „bílá tabule“, která spolu se snímačem a speciálně upravenými popisovači zaznamenává text a případné nákresy. Druhou částí je PC připojené k ovládacímu pultu, které spravuje všechny uvedené prvky a spolu s centrálním serverem záznam ukládá tak, aby bylo možné jednotlivé datové proudy synchronizovat. Po skončení přednášky nastane její post-processing na centrálním serveru, aby bylo možné ji předat poslední části, kterou je webová stránka, v případě TUL na adrese:

<http://prednasky.tul.cz>

Mediasite umístěná a provozovaná na uvedené adrese je ve verzi 5.5.3 a tak je její uživatelská část téměř výhradně tvořena JavaScriptem a technologií Microsoft Silverlight, což zásadně omezuje možnosti jejího provozování víceméně jen na osobní počítače. Aktuální verze Mediasite 7 využívá pro přehrávání videí technologii HTML5, a tak by měla být dostupná i na mobilních zařízeních. Bohužel podle informací správce se aktualizace na vyšší verzi v současnosti neplánuje a proto byl pro splnění zadání zvolený odlišný postup, dále popsany v kapitole 3.4.2. Nová verze Mediasite stojí nemalé finanční prostředky a není ani jasné, zdali by pro streamování obsahu v HTML5 nebylo nutné všechny záznamy převést do jiného datového formátu, což by při současném objemu dat, který každým dnem dále narůstá, bylo velmi náročné.

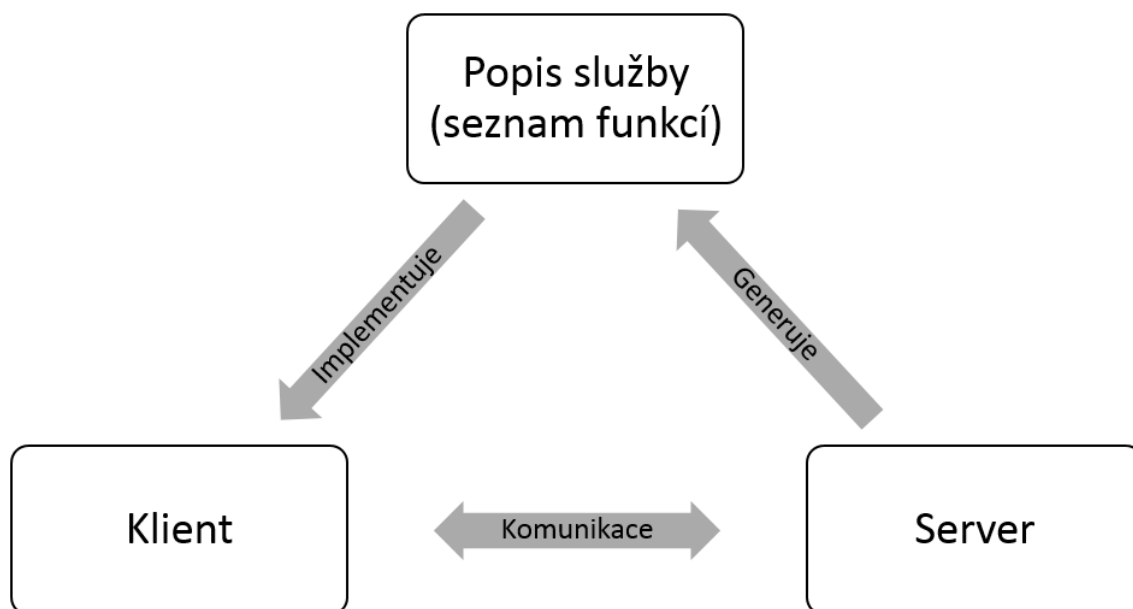
1.2 Bezpečnost

Vzhledem k povaze dat v Moodle a Mediasite musel být kladen velký důraz na celkové zabezpečení aplikací, ale i přenos informací z nich ke klientovi a zpět. Bezpečného přenosu dat je dosaženo protokolem HTTPS. Server při pokusu o přístup na web přes nešifrovaný protokol přesměruje kódem hlavičky *301 Moved Permanently* na zabezpečenou adresu a efektivně tak vynucuje její použití. Nedílnou součástí HTTPS je i certifikát, kterým server prokáže svou identitu a nemůže tak dojít k podvrhnutí, do jisté míry se tak i zamezí *man-in-the-middle* útoku. Mediasite využívá jiných prostředků zabezpečení a snaží se ochranu především uložených děl před jejich neoprávněným stažením nebo sledováním. Dosahuje toho pomocí generovaných tokenů, náhodných cookies a kontrolou IP adresy mezi jednotlivými požadavky. To má zabránit především zneužití díla uživateli například sdílením odkazů. Podrobněji je zabezpečení Mediasite popsáno v kapitole 3.4.2.

Na testovacím serveru bylo nutné napodobit podmínky zabezpečení, aby bylo možné ověřit jeho fungování v praxi s mobilní aplikací. Mediasite je uzavřený projekt, který není volně dostupný a tudíž jej není možné testovat na testovacím serveru. Pro Moodle bylo nutné vytvořit certifikát a nastavit server Apache, aby aktivně vynucoval použití HTTPS. Pro potřeby vývoje není třeba certifikát ověřený uznávanou certifikační autoritou, ale plně postačuje certifikát podepsaný sám sebou, někdy také uváděný pod anglickým označením self-signed.

2. Webové služby a spojení klient-server

Webové služby jsou souborem předpisů standardizující spojení několika systémů. Jsou nedílnou součástí *Service oriented architecture* – architektura orientovaná na služby. Server typicky poskytuje seznam služeb a klient tyto služby volá s parametry a získává tak od serveru data nebo mu je předává. To umožňuje spojení aplikací a informačních systémů bez ohledu na jejich velikost či vnitřní složitost. Velikou výhodou tvoří fakt, že serverová část stačí implementovat jen jednou a případné další aplikace a okolní systémy mohou využívat stávající rozhraní a programátoři na něj mohou napojit své systémy, ryze na základě dokumentace funkcí a datových formátů. Z uvedeného je zřejmé, že webové služby jsou výhradně typu server – klient. V tomto případě jako server figuruje Moodle a jako klient výsledná aplikace.



Obrázek 2.1: Obecné schéma webové služby

2.1 Volba formátu

Moodle již v základní instalaci obsahuje moduly pro webové služby a stačí je v nastavení nakonfigurovat a povolit jejich použití. Implementované formáty jsou následující:

2.1.1 SOAP

Plným jménem *simple object access protocol*. Tento protokol využívá XML pro serializaci dat a jejich přenos přes HTTP/HTTPS a tak bez problému prochází infrastrukturou internetu (např. firewally). Principiálně pracuje tak, že serializuje data a obaluje je vlastní obálkou. Celek potom projde sítí a druhá strana pak data rozbalí z obálky a sestaví zpět původní objekt. V kombinaci s WSDL – specifikací webové služby umožňuje automaticky generovat klientské části zdrojového kódu. Bohužel Moodle neposkytuje kompletní WSDL specifikaci a tak není použití těchto nástrojů možné, jak například uvádí [5]. Po ručním opravení tohoto souboru do použitelného stavu však bylo zjištěno, že současná podoba Silverlight frameworku pro WP8 není kompatibilní s frameworkem *Zend*, konkrétně jeho částí *ZendSOAP*. To je vyvoláno odlišným přístupem ke kódování dat. Zend využívá parametr `use="encoded"`, který Silverlight aktivně odmítá. Postoj společnosti Microsoft k tomuto parametru je značně kritický jak blíže popisuje [6]. Dá se předpokládat, že ani v budoucnu se stav nezmění a tato konkrétní implementace SOAP bude nadále nekompatibilní s Windows Phone. Ačkoli byl tento formát původně vybrán pro implementaci ve vytvářené aplikaci, kvůli uvedeným komplikacím musel být upřednostněn formát jiný.

2.1.2 XML-RPC

Vzdálené volání procedur a využití XML je snadné a standardizované řešení pro webovou službu. Protokol jako takový jen shrnuje soubor pravidel, jak použít konkrétní technologie pro potřeby vzdáleného volání procedur. Data jsou zapouzdřena do XML dokumentu, který je následně odeslán přes HTTP/HTTPS. Výhody i nevýhody má víceméně totožné se SOAP protokolem. Bohužel Moodle má natolik nevhodnou formu XML dokumentů, že není možné s ním využít nativní *serializer* v *.NETu*, což by negativně ovlivnilo přehlednost kódu a také rychlost zpracování větších objemů dat (například pokud by kurz obsahoval velké množství materiálů).

2.1.3 Action message format

Jedná se o binární protokol určený k serializaci objektů a jejich vztahů vyvinutý pro *XML* a *ActionScript* společností Adobe. Je nejčastěji využíván v aplikacích běžících na technologii *Flash*. Vzhledem k tomu, že *.NET* nativně AMF nepodporuje, byl tento protokol vyhodnocen jako nevhodný pro použití ve vyvíjené aplikaci.

2.1.4 REST

Representational state transfer je protokolem, jež se snaží o jednoduché bezstavové rozhraní mezi webovou službou a klientem. Jako přenosové médium využívá HTTP/HTTPS a datový formát nijak pevně nespecifikuje, ačkoli se nejčastěji používá HTML, XML nebo JSON, musí vývojář přímo znát formu dat, aby ji mohl korektně implementovat. Moodle podporuje u tohoto protokolu formát XML a od verze 2.2 také JSON. Ten je rychlostí zpracování obdobně rychlý jako XML. Co se týká datové náročnosti, je na tom dokonce lépe, protože zpráva je kratší. To uživatel ocení v situaci, kdy má velmi pomalé připojení k internetu.

Po zhodnocení všech kladů a záporů uvedených protokolů byl vybrán formát REST pro jeho jednoduchost, přehlednost a možnost kvalitně jej implementovat v prostředí Windows Phone s pomocí nativních nástrojů a jedné externí knihovny. Jak již bylo uvedeno v kapitole 2.1.2, formát dat XML je v Moodle navržen velmi nevhodně, a proto byl upřednostněn JSON.

2.2 Server

Tato kapitola popisuje nutnou konfiguraci systému Moodle pro správný chod klientské aplikace. Názvy položek jsou uvedeny v češtině, tak jak je uvádí český překlad Moodle ve verzi 2.6, který není zcela kompletní a i proto se vyskytují anglická označení. Úpravy jiného než konfiguračního charakteru jsou drženy na minimu. To je dáno jednak velikostí projektu a faktem, že se stále vyvíjí a pravidelně aktualizuje. Nesystematická úprava by v budoucnu mohla poškodit funkčnost jiných komponent. Vyžaduje tedy při aktualizaci zvláštní pozornost a neúměrně by zatěžovala administrátory, což by přidaná hodnota nedovedla vyvážit.

2.2.1 Konfigurace

Systém Moodle, ačkoli moduly pro webové služby obsahuje, nejsou zapnuté a nakonfigurované. Také je třeba upravit přístupová práva pro uživatele. Tyto změny musí provádět administrátor stránek.

- Nejprve je nutné umožnit použití webových služeb a to v menu *Správa stránek* → *Pokročilé funkce* pomocí volby *Povolit webové služby*.
- Dále se musí aktivovat vybraný protokol, v tomto případě REST. To provedeme v menu *Správa stránek* → *Moduly* → *Webové služby* → *Správa protokolů* kliknutím na ikonu přeškrtnutého oka v řádku *REST*.

- Pro autentizaci uživatele musíme definovat novou externí službu. V nabídce *Správa stránek* → *Moduly* → *Webové služby* → *Externí služby* vytvoříme novou službu se jménem *client_windows_phone* a zaškrtneme volby *Povoleno*, *Pouze oprávnění uživatelé*, *Může stahovat soubory*.
- K nově vytvořené službě je třeba přiřadit funkce kliknutím na odpovídající volbu. Je vhodné přidat jen ty, které aplikace využívá a těmi jsou:

```
core_user_get_users_by_field
core_user_update_users
core_enrol_get_users_courses
core_course_get_contents
core_webservice_get_site_info
```

- Službě musí být přidělen parametr *shortname*, to ale není možné udělat z administračního rozhraní. V budoucnu to pravděpodobně možné bude, viz [7]. Je třeba upravit přímo v databázi tabulku *mdl_external_services* a ručně vepsat do pole *shortname* stejný řetězec, jako je v poli *name*, což je v tomto případě *client_windows_phone*.
- Autentizace služeb se provádí pomocí tokenů. Každý uživatel musí mít vygenerovaný unikátní token. Tím je řetězec o délce 32 znaků, který hexadecimálně reprezentuje 64 bajtů čísel. Obsahuje tedy znaky 0–9 a A–F. Vygenerovat jednotlivé tokeny (například pro testovacího uživatele) je možné v nabídce *Správa stránek* → *Moduly* → *Webové služby* → *Správa tokenů* kliknutím na možnost *Přidat*. V případě celoplošného nasazení je ale vhodné využít automatického generování tokenů pro uživatele určité role. V nabídce *Správa stránek* → *Uživatelé* → *Oprávnění* → *Definovat role* u role *Registrovaný uživatel* kliknutím na ikonu ozubeného kola otevřeme nastavení. Zde ve sloupci *Oprávnění* u položek *Generovat token webové služby* a *Vidět skryté detaily uživatelů* zaškrtnutím položky *Povolit* povolíme automatickou tvorbu tokenů a možnost zobrazit detaily uživatele jako je například telefonní číslo.

Tímto je konfigurace systému Moodle dokončena a každý student TUL nyní může využívat aplikaci pro přístup k portálu e-learningu na mobilním telefonu s Windows Phone 8. Systém Mediasite nepotřebuje pro potřeby této práce žádné nastavení, či úpravy.

2.2.2 Vývoj

Při vývoji nastala potřeba testovat webové služby a kontrolovat formát odpovědí, aby mohla aplikace adekvátně reagovat. To plyne především z nekompletního WSDL souboru a velmi skromné dokumentace. Moodle ale poskytuje rozhraní pro testování webových služeb pod položkou *Správa stránek* → *Vývoj* → *Web service test client*. Umožňuje vybrat dvě metody autentizace a to:

- *token* – tato možnost autentizuje funkci s pomocí tokenu, který má uživatel vygenerovaný. Zjistit tento token pro konkrétního uživatele lze pomocí skriptu *token.php* (více viz 2.3.1) nebo, pokud má uživatel token generovaný manuálně, v menu *Správa stránek* → *Moduly* → *Webové služby* → *Správa tokenů*
- *simple* – přihlášení klasicky za použití jména a hesla uživatele, s jehož právy bude spuštěna funkce webové služby

Dále je možné zvolit konkrétní protokol ve stejnojmenném poli. Poslední položkou je výběr funkce webové služby. Po potvrzení výběru se provede autentizace dotazem na token nebo jméno a heslo, zvolená funkce se provede a její výsledek je vypsán na obrazovku.

Seznam funkcí v testovacím rozhraní obsahuje jen několik základních (a *deprecated*) funkcí a proto postačuje jen k základní představě o fungování služeb a funkcí. K ručnímu testování byla využita služba <http://apikitchen.com>, která umožňuje pohodlnou a uživatelsky přívětivou tvorbu požadavků na libovolný server, ať už

webových služeb nebo jiných aplikačních rozhraní, pokud využívají pro svůj přenos HTTP/HTTPS protokolu a jsou dostupné ze sítě Internet.

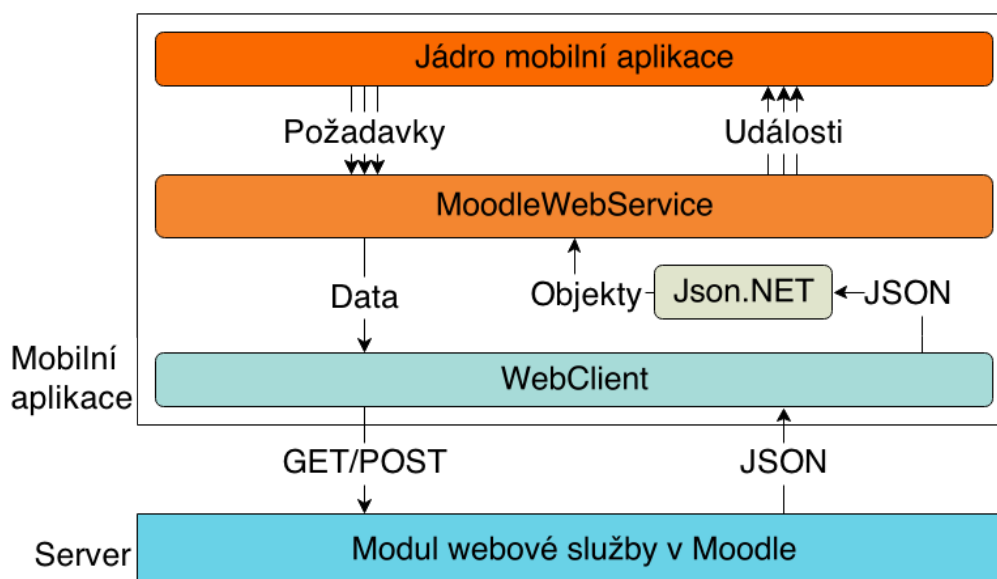
Při implementaci funkce *core_user_update_users* vyšlo najevo, že ve zdrojovém kódu se kontroluje oprávnění *moodle/user:update*. To řadový uživatel běžně nemá, protože umožňuje editaci všech uživatelů. Pokud se uživatel pokouší o úpravu vlastního profilu, bylo by správněji kontrolovat oprávnění *moodle/user:editownprofile*, které mu to umožňuje a na TUL je vlastní každý student. Tato chyba je již ohlášena na oficiálním webu jako CONTRIB-4282 [8] i s navrhovaným řešením, které bylo na testovacím serveru vřazeno do kódu. Dá se očekávat, že budoucí verze již budou mít tuto chybu opravenou a tak je tato manuální oprava víceméně dočasným řešením.

Při ukládání dat bylo také zjištěno, že implementace webové služby z neznámých důvodů neumožňuje editaci dodatečných údajů jako jsou telefon, adresa a jiné. Ačkoli tyto nejsou nijak klíčové, je vhodné možnost jejich editace zpřístupnit a proto byl editován soubor */user/externallib.php*, konkrétně přidáním těchto hodnot ve funkci *update_users_parameters*. Bez této editace by uživatel nemohl v podstatě upravovat nic, neboť základní údaje jako jméno, příjmení, e-mail atd. jsou načítány z centrálního serveru identit Liane a Moodle má práva pouze ke čtení těchto dat.

2.3 Klient

Komunikaci mobilní aplikace s webovou službou zajišťuje nově vytvořená třída *MoodleWebService*. Poskytuje sadu funkcí a událostí pro práci s webovou službou běžící na e-learningovém portálu a implementuje ověření identity uživatele. Třída je označena jako *partial* a je tak pro přehlednost rozdělena do více souborů. To umožňuje dodržet konzistentnost „jedna služba – jedna třída“ a zároveň přehlednost, která je velmi důležitá. Jako celek udržuje důležité adresy a formát dotazů na server a dalším částem aplikace poskytuje jednoduché a intuitivní rozhraní. Součástí jmenného prostoru *MoodleWS* jsou také třídy, které mají příslušné atributy a třída

Json na jejich základě zpracuje odpovědi serveru a vytvoří daný objekt. Komunikace probíhá přes Internet po zabezpečeném protokolu HTTPS, který e-learnigový server aktivně vynucuje. Na základě uvedeného přirozeně vzniklo logické rozdělení klienta webové služby uvedené na obrázku.



Obrázek 2.2: Hlavní části aplikace

2.3.1 Autentizace a token

Autentizace uživatele a získání tokenu se liší od použití webových služeb a je pro všechny společné. Uživatel je ověřován za pomoci jména a hesla. V případě TUL jsou tyto údaje shodné s přístupovými údaji do sítě LIANE.

Samotné ověření se provádí skriptem */login/token.php* metodou GET s parametry pojmenovanými *username*, *password* a *service*. Třetí uvedený parametr je krátké jméno služby, Moodle označované jako *shortname* a určuje tak, ke které službě se klient přihlašuje. Toto jméno se musí shodovat se jménem uvedeným v databázi, pro potřeby této aplikace bylo použito *client_windows_phone*, viz kapitolu 2.2.1. Vzorový požadavek odesílaný na server pak vypadá například takto:


```
https://elearning.tul.cz/login/token.php?username=martin.pomezny&password=tajemstvi&service=client_windows_phone
```

Server pak odpovídá ve formátu JSON, bez ohledu na to, jaký protokol využívá daná externí služba. Pokud při přihlašování nastane chyba, například uživatel zadá špatné heslo nebo administrátor vypne externí službu, odpoví script chybovým hlášením. V případě nesprávných přihlašovacích údajů takto:

```
{"error": "Uživatelské jméno nebylo nalezeno v~databázi.",  
"stacktrace": null,"debuginfo": null,"reproductionlink": null}
```

Parametry *stacktrace* a *debuginfo* jsou null, protože je z bezpečnostních důvodů vypnuté zobrazování ladících informací. Pokud by bylo zapnuté, obsahovaly by detailní informace o chybě a o tom, kde nastala. Aplikace však přesto dále předá formou výjimky jen zprávu v parametru *error*. Úspěšné přihlášení je signalizováno tím, že server poskytne token:

```
{ "token": "ba1e917744a17f5b8ff9319c92fc287e" }
```

Token je ukládán jen dočasně pro jedno přihlášení a po každém zapnutí aplikace je znovu stahován ze serveru. Toto pravidlo je zavedeno záměrně, protože umožňuje implementovat lepší zabezpečení, více v kapitole 3.7. K dočasnému uložení tokenu je použito izolované úložiště *Isolated Storage*, které lze považovat za bezpečné. Více o něm a jeho obsluze v kapitole 3.1.

Vzhledem k bezpečnosti uživatelů je nutné zmínit, že v základním nastavení se na serveru Apache do logů ukládají požadavky typu GET v celé své délce, což může v případě napadení serveru a zpřístupnění logů útočníkovi vést k vyzrazení hesel jednotlivých uživatelů. Bohužel Moodle neumožňuje získat token jinou cestou a tak je vhodné upravit nastavení serveru Apache tak, aby tyto požadavky nezapisoval do provozních záznamů.

2.3.2 Framework Json.NET

Před samotnou implementací jednotlivých funkcí webové služby bylo nutné zajistit rychlý a spolehlivý způsob jak z dat ve formátu JSON vytvořit jednotlivé instance tříd. Po zvažování výhod a nevýhod byl zvolen framework *Json.NET*, který je podle jeho autora rychlejší až o 50 % než nativní *DataContractJsonSerializer*, který je součástí frameworku *.NET* [9]. Taktéž v případě rozšíření aplikace poskytuje zhruba trojnásobek funkcí a konfiguračních možností.

Framework tvoří knihovna distribuovaná pod licencí MIT [10] z roku 2007. Jde tedy integrovat i do komerčních projektů zcela zdarma. Vřazena do projektu byla z repozitáře *NuGET* přímo ve vývojovém prostředí Visual Studio 2012 a je součástí projektu ve verzi 5.0.8. V době psaní této práce se rozšiřoval seznam plánovaných vylepšení a zdrojový kód na adrese <https://github.com/JamesNK/Newtonsoft.Json> se stále intenzivně vyvíjel. Vzhledem k licenci pod kterou je knihovna vydaná a velkému zájmu uživatelů se dá předpokládat, že její vývoj bude dále pokračovat a bude i nadále použitelná, což je výhodou, pokud by se vyvíjená aplikace v budoucnu upravovala pro novější verzi Windows Phone nebo se přistoupilo k implementaci dalších funkcí.

Správné deserializace objektů je dosaženo s pomocí atributů frameworku. Samotná třída je označena jako *[JsonObject]*, jednotlivé členské proměnné pak *[JsonProperty("jmeno")]*. Tím je možné z názvů které používá Moodle (vše malými písmeny) přejít na názvy v tzv. *PascalCase*, který je zavedenou konvencí CLR jazyků *.NET*.

2.3.3 Funkce

Volání jednotlivých funkcí webové služby zajišťuje vytvořená třída *MoodleWebService* ve jmenném prostoru *MoodleWS* společně s třídami odpovědí v *MoodleWS.Responses*. Kdyby nedošlo ke komplikacím uvedeným v 2.1.1 a byl by použit

SOAP protokol, bylo by možné tyto třídy vygenerovat automaticky pomocí utility *SLsvcUtil.exe* nebo volbou „Add Service Reference“ ve vývojovém prostředí Visual Studio 2012.

Třída každou funkci webové služby reprezentuje jako veřejnou metodu ve tvaru *JmenoFunkceAsync*, což odpovídá zavedeným konvencím v jazyce C#. Slovo *Async* na konci funkce dává programátorovi jednoznačně najevo, že se jedná o asynchronní akci a před jejím zavoláním by měl registrovat funkci jež bude reagovat na vyvolanou událost po skončení akce. Metody přijímají různé parametry a na jejich základě vrací v parametrech **AsyncCompletedEventArgs* příslušné datové objekty nebo v poli *Error* vyvolanou výjimku.

- *GetTokenAsync(string username, string password)* – tato metoda se pokusí získat token za pomoci předaného jména a hesla. *Shortname* služby čerpá z konstanty uložené uvnitř třídy. Pokud se autentizace nezdaří, předá se chybové hlášení uvnitř výjimky v poli *Error* objektu *GetTokenAsyncCompletedEventArgs*. Pakliže se přihlášení vydaří a server vrátí platný token, nabývá pole *Error* hodnoty null a token je předán jako string v poli *Token*. Programátor má pak možnost uložit token do izolovaného úložiště a předávat jej odtamtud nově vytvářeným instancím třídy *MoodleWebService* a nebo jej nastavit již vytvořené instanci jako pole token.

Následující metody vždy před svým samotným kódem volají metodu *CheckToken()*. Ta ověřuje, že instance má nastavený token a může jej tak přidat k volání funkce webové služby. Absence tokenu nebo jeho neplatnost vyvolá výjimku. Tato situace by ovšem neměla nastat, protože programátor by měl volat funkce jen pro autentizované uživatele.

- *GetSiteInfoAsync()* – zajistí stažení informací o serveru. Z hlediska aplikace jsou důležité především atributy které se týkají uživatele přihlášeného pomocí odeslaného tokenu. Jsou jimi jméno, příjmení, uživatelské jméno a id uživatele. Poslední uvedené je nezbytně nutné a proto je pole *UserId* označeno

jako *Required* pro deserializaci frameworkem *Json.NET*. Odpovídající funkcí webové služby je *core_webservice_get_site_info*.

- *GetCoursesAsync(int userId)* – metodě je předáno parametrem unikátní id uživatele, získané například voláním předešlé metody. Návrátovou hodnotou je *List<Course>* předaný v poli *Courses* parametru typu *GetCoursesAsyncCompletedEventArgs*. Jedná se o seznam kurzů, do kterých je uživatel s daným identifikátorem zapsán. Kurz je reprezentován třídou *Course*, obsahuje krátké a dlouhé jméno a jedinečný identifikátor kurzu. Seznam kurzů a data jsou získána pomocí funkce *core_enrol_get_users_courses*.
- *GetCourseContentAsync(int courseId)* – tato metoda načte seznam jednotlivých sekcí v konkrétním kurzu určeném parametrem *courseId* včetně obsahu sekcí. Využívá se funkce *core_course_get_contents*. Kurz je rozdělen do sekcí, které obsahují moduly a v těch jsou příslušné obsahy. Tyto tři klíčové prvky reprezentují třídy *CourseSection*, *CourseModule* a *ModuleContent*. Jako sekce bývají použity jednotlivé týdny kalendáře, moduly jsou konkrétní materiály. V obsahu modulu je odkaz a data o materiálu. Ačkoli Moodle umožňuje do jednoho modulu přidat více obsahu a webová služba jej korektně vrátí, webové rozhraní zobrazí pouze první vloženou položku obsahu. Pokud tedy vyučující například přetáhne do okna více obrázků, studenti vidí pouze první z nich. Třída *CourseModule* implementuje stejné chování, aby byl přístup k materiálům konzistentní s webovým rozhraním.
- *GetUserProfileAsync(int userId)* – kompletní profil přihlášeného uživatele je získán využitím funkce *core_user_get_users_by_field*. Ta umožňuje získat od webové služby profil uživatele podle různých unikátních polí. V tomto případě je použito id uživatele a funkce je tak využita obdobně jako *core_user_get_users_by_id*, která je však již označena *deprecated* a neměla by se tak v nových projektech používat. Funkce očekává na vstupu pole a tudíž je jí předáváno podle s jedním prvkem. Profil je poskytován jako instance třídy *UserProfile*.

- *UpdateUserProfileAsync(UserProfile profile)* – metoda, která s pomocí funkce *core_user_update_users* aktualizuje uživatelský profil na serveru. Pro potřeby TUL je upravena a neaktualizuje základní atributy jako jméno, příjmení, e-mailovou adresu, login a heslo. Tyto se načítají z centrální databáze LIANE a k jejich editaci by tedy muselo docházet tam. Moodle ale z bezpečnostních důvodů přistupuje k tomuto zdroji s přístupovými právy pouze pro čtení.

3. Aplikace

Nezbytnou částí této práce bylo vyvinout samostatnou mobilní aplikaci pro systém Windows Phone 8. Tato aplikace bude sloužit studentům pro připojení k e-learningovému portálu Moodle a přístupu k materiálům, jako jsou přednášky nebo jiné dokumenty vložené vyučujícím. Aplikace by měla mít intuitivní ovládání a preferovat přehlednost. To, že se graficky nepodobá samotnému portálu, není překážkou, protože ctí všechna doporučení a zásady pro tvorbu aplikací pod WP8. Ovládání je tak velmi podobné ostatním aplikacím celé platformy a připadá tak uživateli velmi přirozené a intuitivní.

Cílem bylo vytvořit konzumní aplikaci, jež umožní obsah stahovat a zobrazovat. Tvorba obsahu a nahrávání souborů ustoupily do pozadí, což je způsobeno i faktem, že celý systém není k tomuto uzpůsoben a uživatel například nemá přístup k souborovému systému a nemůže tedy nahrávat některé soubory jiných aplikací. Typickým použitím je nahlédnutí například do prezentace z přednášky při cestování nebo diskusi u oběda v menze.

Pro samotný vývoj aplikace byla důležitá volba integrovaného vývojového prostředí (IDE). Tím bylo *Microsoft Visual Studio Express 2012 for Windows Phone*. Je poskytováno zcela zdarma a obsahuje všechny nezbytné knihovny a součásti pro vývoj aplikací. Součástí SDK je i emulátor s několika různými verzemi systému lišících se především rozlišením displaye a velikostí operační paměti. Je možné jej využít i k testování polohových služeb a omezení rychlosti připojení k síti. Ačkoli je emulátor výhodný pro zevrubné otestování na různých konfiguracích je vhodné

otestovat aplikaci i na skutečném zařízení. K tomuto účelu byl použit mobilní telefon Nokia Lumia 920. I proto by aplikace neměla mít problém v přijetí na Windows Phone Store, až budou provedeny změny na serveru a rozhodnuto o jejím uvolnění mezi studenty.

Aplikace umožňuje studentovi základní práci s portálem z prostředí jeho telefonu. Portál jako takový obsahuje mnohem větší škálu funkcí a činností. Implementace všech by však byla nad rámec této práce, pro některé z nich ani nejsou připravené webové služby v Moodle a některé by z hlediska použití na mobilním telefonu byly naprosto zbytečné. Aplikace tedy implementuje následující funkce:

- Přihlášení a editace uživatelského profilu.
- Zobrazení kurzů a jejich obsahu.
- Registrace kurzů podle STAGu.
- Možnost otevírat a stahovat nahrané materiály.
- Přehrávání streamů ze serveru Mediasite.

3.1 Izolované úložiště

Většina aplikací má potřebu ukládat si mezi jednotlivými spuštěními nějaká data. Těmi mohou být konfigurační data nebo celé datové soubory. To v systému Windows Phone zastává úložiště *Isolated storage*. Každá aplikace má svůj prostor, který je chráněn proti přístupu z ostatních aplikací. Přístup k těmto datům je možný připojením telefonu k počítači a použitím nástroje *ISETool.exe*, to ovšem platí jen pro aplikace instalované ve vývojovém režimu (někdy označované jako *sideloaded*). Pokud je aplikace stažena a nainstalována z obchodu Windows Phone Store je digitálně podepsaná a telefon přístup touto cestou odmítne. Vstup do úložiště povolí jen aplikaci s odpovídajícím identifikátorem a podpisem. Tím je dosaženo

dostatečného zabezpečení dat a může zde být uloženo jméno a heslo, pokud si to uživatel přeje.

Základem jsou dva přístupy k úložišti. Prvním je přístup ukládání celých souborů. Tento způsob je vhodné použít například pro cache obsahu stahovaného ze sítě, pracovní soubory, logy, mapové podklady atd. Aplikace pak má složku se soubory a práce s nimi je čistě v její režii. Druhou možností je využít *IsolatedStorageSettings*. To, jak již napovídá název, je vhodné především pro ukládání nastavení aplikace a využívá strukturu podobnou slovníku pro serializované objekty.

Pro snazší práci s *IsolatedStorageSettings* byla vytvořena třída *AppSettings*, která je zapouzdřuje a ošetřuje chybové stavy. Například při pokusu o čtení vlastnosti, která ještě nebyla uložena je podstrčena předem nastavená hodnota (tzv. *default*). Tento stav běžně nastává při prvním spuštění aplikace a díky této implementaci nemusí hlavní část programu tyto stavy ošetřovat a může spolehlivě pracovat jen s operací přiřazení.

3.2 Zápis předmětů ze STAGu

Technická univerzita využívá ke kompletní evidenci studentské agendy systém STAG. Ten eviduje v podstatě všechna důležitá data týkající se studentů, jejich hodnocení, stav kreditů, rozvrhy a také kvalifikační práce. E-learningový portál je s ním propojen a umožňuje se studentům pohodlně zapisovat do kurzů (v terminologii STAG předmětů). Příslušným předmětům ve STAGu jsou vytvořeny jejich protějšky v Moodle a uživatelé jsou pak nabídnuty kurzy, které podle STAGu studuje. Student vybere do jakých kurzů by se rád zapsal a potvrdí formulář. Tím se mu v Moodle označené kurzy zapíší. Alternativně může studenty ručně přidat vyučující, to je ale zbytečně náročné a je vhodné, když se studenti zapisují sami. Do kurzů se nelze zapsat přímo a i proto byla tato funkce zařazena do aplikace.

Zápis předmětů ze STAGu není standardní součástí systému Moodle, ale je řešen jako modul, který pro TUL vytvořil Ing. Igor Kopetschke. Vzhledem k tomu, že modul neimplementuje funkce pro volání způsobem webových služeb, musely být principy jeho fungování ručně zjištěny a obstarány v aplikaci jiným způsobem.

Stažení informace o dostupných kurzech k zapsání se realizuje přímo voláním modulu, který vrátí HTML stránku. Autentizace je zajištěna předáním tokenu v GET parametru *token*. Výsledný požadavek odesílaný na server vypadá například takto:

```
https://elearning.tul.cz/mod/stag/view.php?  
token=ba1e917744a17f5b8ff9319c92fc287e
```

Na stránce se nachází formulář obsahující jednotlivé položky k zapsání jako html element `<input type="checkbox">`. Popisky jednotlivých elementů a jejich hodnoty jsou v HTML kódu vyseparovány za pomoci dvojice regulárních výrazů. Z těchto dat jsou utvořeny vizuální komponenty a jako součást aplikace prezentovány uživateli. Ten zatržením vybere, které kurzy si zapíše a volbu potvrdí. Aplikace následně odešle POST data obsahující označené kurzy na stejnou adresu a se stejným tokenem. Tím se de facto emuluje odeslání formuláře uživatelem z webu, na což modul reaguje provedením patřičných úkonů.

Touto odchylkou od ostatních funkcí na klientské straně bylo dosaženo plné funkčnosti v mobilní aplikaci, aniž by se modul na serveru musel upravovat. Provedení je konzistentní se zbytkem aplikace a uživatel nemá šanci poznat, že je něco jinak.

3.3 Komunikace se serverem

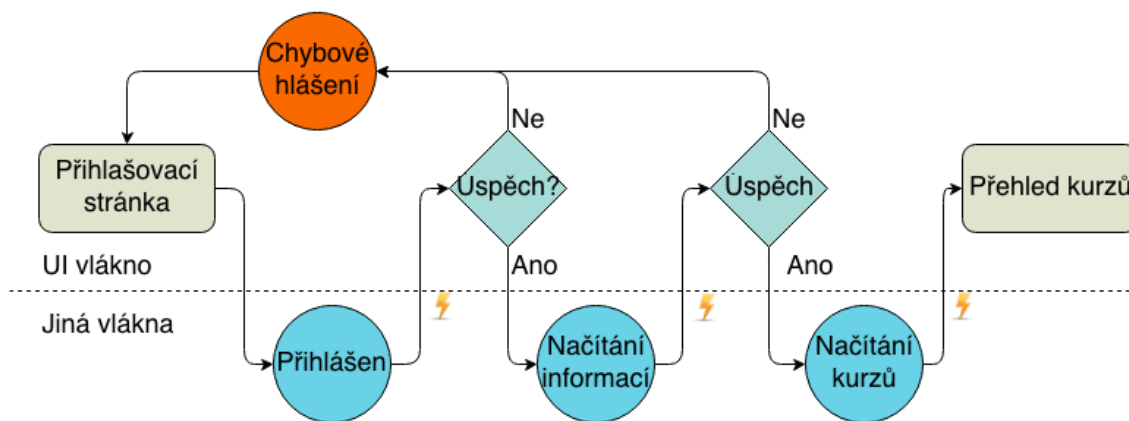
Spojení se serverem je realizováno skrze síť Internet – po zabezpečeném protokolu HTTPS. Jak dlouho bude trvat, než se odpověď na zasláný dotaz dostane do zařízení,

není možné předem určit či odhadnout, neboť závisí na množství faktorů, mezi které patří rychlost odesílání a stahování dat, momentální zátěž serveru atd. Navíc lze předpokládat, že komunikace se serverem může být i velmi pomalá a tak může provedení požadavku trvat i několik desítek sekund. S tím by se mohli potýkat především uživatelé bez vysokorychlostního připojení k internetu a nebo při omezení operátorem po překročení svého datového limitu.

V případě delší prodlevy musí být uživatel informován o probíhající komunikaci a aplikace se nesmí zaseknout v čekání. Uživatel musí mít pocit, že aplikace stále plynule reaguje na jeho podněty a také musí mít možnost probíhající požadavek zrušit návratem na předchozí stránku aplikace.

Signalizace probíhající komunikace je realizována komponentou *ProgressIndicator*, která při nastavení *IsIndeterminate = true* zobrazuje uživateli, že operace probíhá, ale neukazuje její postup a kolik zbývá do konce operace. To je v případě požadavku na webovou službu příhodné, neboť jak již bylo uvedeno, časovou náročnost nelze určit. Komponenta je standardní součástí frameworku a využívá se ve většině aplikací pro zobrazení probíhající operace. Je umístěna u horního okraje obrazovky a její vizuální reprezentace sestává z názvu operace a 5 teček v barvě tématu telefonu přijíždějících z levé strany, zpomalujících u středu a následně odjíždějících vpravo.

Aby aplikace při čekání na dokončení provedení operace byla schopna reagovat na povely uživatele, nesmí danou operaci obsluhovat vlákno pro uživatelské rozhraní, takzvaný *UI thread*. Namísto toho musí být operace asynchronně obsloužena a na pozadí vytvořeno nové vlákno, které vyčká na dokončení a pak formou události předá data zpátky hlavnímu, v tomto případě UI vláknu.



Obrázek 3.1: Ukázka asynchronního průběhu přihlášení

Protože zvolený protokol využívá standardního přenosu po HTTP, mohla být využita třída *WebClient*, která je pro tyto operace určená. Tu využívá třída *ModuleWebService* popsaná v kapitole 2.3.3 k odesílání a stahování dat asynchronním způsobem a to tak, že pomocí privátních metod zařídí naformátování dat a připojení předaného event handleru k události příslušné instance třídy *WebClient*. Metody se liší jen tím, jak s danými daty zachází. Vstupní parametry mají stejné. Parametr *address* obsahuje adresu, kam budou daná data odeslána, *data* je slovník obsahující dvojice „jméno - hodnota“. Zpracovává se tak, že pro každou položku slovníku se spojí jméno a hodnota pomocí znaku „=“ a jednotlivé položky takto utvořené se spojí znakem „&“. *EventHandler* je funkce, jež se spustí po přijetí dat.

- *SendGet(string address, Dictionary<string, string> data, DownloadStringCompletedEventHandler eventHandler)* – tato metoda data odesílá na zadanou adresu metodou GET. Jediné, co doplní, je kódování dat do hlavičky požadavku a znak „?“ mezi adresu a datový řetězec. K odeslání se používá metoda *DownloadStringAsync* třídy *WebClient*.
- *SendPost(string address, Dictionary<string, string> data, UploadStringCompletedEventHandler eventHandler)* – metoda odesílá data formou POST požadavku, používá tedy metodu *UploadStringAsync*.

- *SendJsonREST(string address, Dictionary<string, string> data, DownloadStringCompletedEventHandler eventHandler)* – tato metoda je určená k odesílání požadavků na webovou službu, funguje obdobně jako první uvedená metoda, jen s tím rozdílem, že k odesílaným datům přidává dvojici *moodlewsrestformat=json*.

Při odesílání požadavku na webovou službu je nutné přidat k adrese GET parametr se jménem *moodlewsrestformat* s hodnotou *json*. To zajistí, že webová služba zašle odpověď ve formátu JSON. Pokud by parametr nebyl uveden nebo se hodnota lišila, odpovídala by služba formátem XML.

3.4 Přehrávání z Mediasite

Záznamy přednášek z Mediasite jsou v Moodle prezentovány jako odkazy mimo web e-learningu. V repozitáři modulů pro Moodle sice existuje modul pro propojení s technologií Mediasite, ale na portálu TUL nasazen není. Místo něj je použit vlastní modul, který odkáže na samotný Mediasite a další činnost studenta probíhá přímo v něm.

E-learningový portál TUL funguje na starší verzi Mediasite (konkrétně 5.5) a tak umožňuje přehrávání jen v technologii Silverlight společnosti Microsoft. Na osobních počítačích funguje aplikace Mediasite bez problémů na všech 3 majoritních systémech. Na mobilních zařízeních je však situace odlišná a ani nejnovější systém pro mobilní telefony, kterým je v době tvorby této práce Windows Phone 8, nedokáže kompletně s webovými applety Silverlight spolehlivě spolupracovat.

Aby byla uživateli nabídnuta možnost využívat multimediální portál Mediasite, byl učiněn ústupek a do aplikace byla zařazena funkce přehrávání videozáznamu mimo Silverlight přímo v telefonu. Důvodem byl fakt, že přednášky jsou dostupné pouze na Mediasite, kdežto například prezentace nebo jiné podklady mohou být, a mnohdy také bývají, uloženy v Moodle.

Samotné přehrávání datového proudu s videem netvoří žádnou překážku, protože je použit kodek WMV3 a ten zařízení s Windows Phone 8 bez problému přehrávají. Překážkou je adresa proudu, kterou Mediasite neprezentuje přímo, ale s ochranou tokeny¹ a cookies ji „schovává“ v souboru JavaScriptu. K dosažení tohoto cíle byl tedy zvolen netypický přístup, kterým byla i částečně zdokumentována ochrana kterou Mediasite implementuje.

3.4.1 Zabezpečení

Portál Mediasite implementuje dvě základní ochrany. První je zabezpečení dat před uživateli, kteří nemají do systému přístup. Toho dosahuje pomocí jména a hesla. Pokud útočník nemá heslo, nedostane se do zabezpečené části webu a nezná tak adresu podepsanou tokenem, ergo nemůže zdroje využívat. Druhá implementovaná ochrana má za cíl zamezit sdílení zdrojů uživateli, kteří přístup mají, ale poskytl by jej i těm, kteří jej mít nemají, například zasláním odkazu na internetovou stránku. Text se dále zabývá touto variantou, protože pokud má student v Moodle přednášku uvedenou, je nade vší pochybnost, že k ní má oprávněný přístup.

Zpřístupnění materiálu skrz Moodle funguje obdobně u obou modulů (tj. oficiální modul v repozitáři i autorský modul TUL). V Mediasite je vytvořen uživatel, který má přístup ke všem zdrojům a v Moodle jsou uvedeny jeho přístupové údaje v konfiguraci modulu. Pokud uživatel v Moodle klikne na přednášku, je přesměrován na portál Mediasite, kde je jednorázově autorizován tokenem v GET parametru. Z pohledu uživatele vše funguje velmi sympaticky, na pozadí se však stane celá řada věcí. Po kliknutí se modul z Moodle pomocí zadaných přístupových údajů přihlásí do Mediasite, kde si nechá vygenerovat odkaz na materiál, který student chce vidět. Tuto adresu potom podstrčí do vytvořeného webu a uživatele na ni přesměruje.

¹Tokenem je v této kapitole označován obecný, jednorázový, kód který autorizuje uživatele pro přístup k nějakým zdrojům. Neplatí pro něj nutně fakta uvedená mimo tuto kapitolu, protože se jedná o token zcela jiné služby.

U webových služeb se situace liší jen v tom, že adresa není nikam podstrkávána a přesměrována, ale je předána službou přímo jako odkaz ve tvaru:

```
http://prednasky.tul.cz/TUL/Viewer/?  
peid=d3fbeaf728144446adffef8f345b1e561d&  
authTicket=a1f048dffa046ec8a53f16f15800ac7
```

Tento odkaz není chráněn před přeposláním, což je pochopitelné, protože kdyby byl chráněn, uživatel by jej nezobrazil (byl vygenerován pro Moodle a jiný počítač s jinou IP adresou). Dá se předpokládat, že tohoto chování je dosaženo parametrem *authTicket*, po jehož odstranění odkaz přestane fungovat a stejně tak se bez tohoto parametru generují odkazy pro uživatele při přístupu přímo na web *prednasky.tul.cz*.

K samostatnému zabezpečení proudu dochází tak, že při přístupu na adresu uvedenou výše, je v HTML kódu stránky uveden odkaz na JavaScript s jiným tokenem, který se po každém načtení webu liší od předchozího a je tedy jednorázový. Odkaz vypadá například takto:

```
http://prednasky.tul.cz/TUL/FileServer/Presentation/  
d3fbeaf728144446adffef8f345b1e561d/manifest.js?  
playbackTicket=3faba0596c1643e29c459cccca139ed0
```

Tento script je již chráněn. Aby se jeho obsah načetl, je nutné aby klient odeslal cookie se jménem *ASP.NET_SessionId* s hodnotou, která odpovídá hodnotě, kterou do cookie vložil server při načtení celého webu prvním požadavkem. Cookie je navíc při ukládání serverem označena jako *HttpOnly*, což má za následek omezení přístupu (podrobněji je práce s ní popsána v následující kapitole). Pokud cookie není v požadavku uvedena nebo její hodnota nesouhlasí s hodnotou uloženou na serveru, je v obsahu souboru jen strohý kód v jazyce JavaScript s hlášením:

```
ErrorStatus = function(){};ErrorStatus.Code=403;ErrorStatus.Text=  
'You are not authorized to view this resource.';
```

Pokud se hodnoty shodují, je v souboru relativně dlouhý kód v jazyce Javascript obsluhující přehrávání. Jeho obsah je podle všeho licenčně chráněn a bez výslovného písemného souhlasu Sonic Foundry jej nelze reprodukovat, měnit, šířit či kopírovat a proto nemůže být uvedena ukázka.

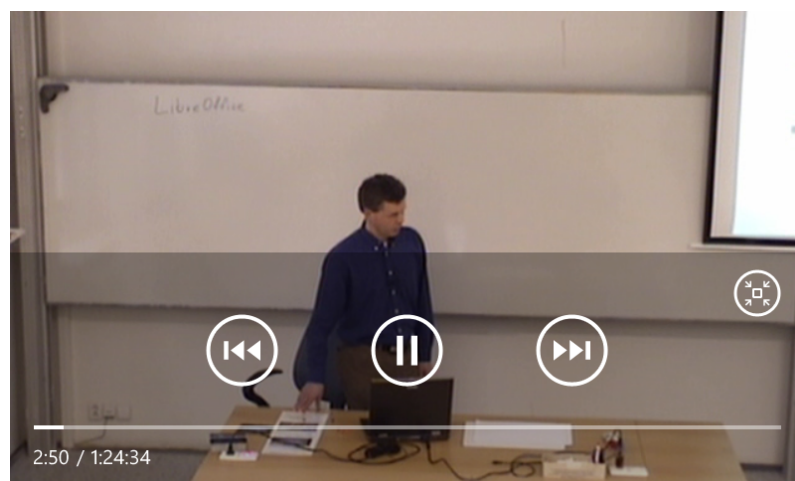
V souboru manifest je v konfigurační části (kolem řádku č. 85) uveden i odkaz na datový proud s videem ve formě odkazu protokolu *MMS*. Součástí adresy je opět token, autentizující uživatele, který zajistí, že server začne streamovat data. Tento odkaz již není chráněn pomocí cookies, protože lze přehrát v libovolném přehrávači podporující daný kodek, kterým je v tomto případě WMV3. Funkčnost byla otestována svobodným přehrávačem *VLC media player*. Odkaz však nezůstává bez ochrany a lze přehrát jen pokud se IP adresa z pohledu streamovacího serveru shoduje s IP adresou, která si vyžádala *manifest.js*.

Adresy byly v této kapitole uměle odřádkovány pro větší přehlednost, ve skutečnosti neobsahují žádné znaky pro nový řádek.

3.4.2 Přehrávání v aplikaci

Pokud uživatel při prohlížení seznamu materiálů v kurzu klikne na materiál s typem „mediasite“, aplikace tento typ zpracuje a spustí se přehrávač integrovaný v prostředí telefonu, na který je uživatel zvyklý, protože jej využívá většina aplikací pro přehrávání videa.

Protože třída *WebService* nepracuje s cookies, bylo nutné třídu zdědit a vytvořit novou třídu *CookieAwareWebClient*, která pomocí třídy *CookieContainer* přidává funkcionalitu uchování a odeslání cookies proměnných, pokud jsou provedeny požadavky jednou instancí vytvořené třídy. Protože cookie *ASP.NET_SessionId* je označena jako *HttpOnly*, není možné k její hodnotě v programu přistoupit a musí se použít kontejner, který ji uchová a programátorovi zůstane skryta.



Obrázek 3.2: Ukázka přehrávače. Ovládací prvky se po poklepnání schovají.

Po kliknutí je potřeba provést dva požadavky a jejich zpracování programem, čehož je dosaženo opět asynchronně jako u obsluhy webové služby a to pomocí třídy *CookieAwareWebClient*. Prvním požadavkem se stáhne obsah titulní strany Mediasite s přehrávačem technologie Silverlight. V aplikaci se po dokončení stahování obsahu vyvolá metoda *MediasiteDownloaded*. Ta ověří, zdali nedošlo k chybě a pokud je vše v pořádku, najde pomocí regulárního výrazu místo s odkazem na soubor *manifest.js*. Ten je stažen pomocí stejné instance třídy *CookieAwareWebClient* jako předchozí. Po dokončení přenosu metoda *ManifestJSDownloaded* ověří, zdali nedošlo k chybě nebo server neodpověděl chybovým hlášením a posléze pomocí regulárního výrazu najde odkaz na datový proud. Odkaz je ve formě URI a je nutné schéma *mms* nahradit schématem *http*. Nyní je možné vytvořit instanci třídy *MediaPlayerLauncher*, předat jí URI v této podobě, nastavit dodatečné parametry jako například orientaci obrazovky a zavoláním metody *Show* spustit přehrávání v přehrávači.

3.5 Navigace

Systém Windows Phone využívá koncept ploché navigace. Jak uživatel pracuje s telefonem, systém si pamatuje jeho poslední kroky a umožňuje se mu vracet s pomocí hw tlačítka o krok zpátky, případně jej může podržet a vrátit se do jiné aplikace.

Aplikace jsou zpravidla tvořeny ze stránek, mezi kterými se uživatel pohybuje. Pohyb zajišťuje služba *NavigationService*, která je členskou proměnnou zděděnou od třídy *PhoneApplicationPage*. S využitím této služby má programátor možnost navigovat uživatele po aplikaci. Mezi její klíčové metody a proměnné patří:

- *Navigate(Uri source)* – tato metoda umožňuje otevřít novou stránku přes stávající. Stávající stránka se drží v paměti a systém se sám postará o zařazení do fronty, aby při stisku tlačítka zpět byla obnovena původní stránka. Předává se jí instance třídy *Uri*. Ta je tvořena jako relativní odkaz přímo na xaml soubor v adresářové struktuře aplikace.
- *CanGoBack* – členská proměnná typu *bool*. Sděluje, zdali je možné se pohybovat zpět v navigačním řetězci.
- *GoBack()* – Vráti se na předchozí stránku, které se získá tak, že se odebere vrchol ze zásobníku *BackStack*. Pokud není v třídě stránky přetížena metoda *OnBackKeyPress*, chová se stisknutí tlačítka „zpět“ obdobně jako volání této metody.
- *BackStack* – v této proměnné je uchován postup, jak se uživatel pohyboval aplikací. Zásobník má na dně vstup do aplikace a na vrcholu poslední stránku. Při návratu na předchozí se odebírají stránky z vrcholu.
- *RemoveBackEntry()* – metoda odebírá vrchol zásobníku, aniž by kamkoli navigovala. Její použití je vhodné, pokud uživateli chceme zabránit v návratu na předchozí stránku, například na přihlašovací formulář.

Mezi stránkami je možné předávat data. Pokud jde o komplexní objekty, je vhodné využít například úložiště *Isolated storage* a metod *OnNavigatedFrom* pro uložení dat a *OnNavigatedTo* pro načtení a přiřazení. Pokud stačí předávat textový řetězec, je možné využít přímo vlastnosti URI a připojit jej za adresu cílové stránky ve tvaru *?klíč=hodnota*. Tímto způsobem je předáván například unikátní identifikátor kurzu stránce s výpisem jeho obsahu. Příkaz pro otevření stránky vypadá takto:

```
NavigationService.Navigate(new Uri("/CourseContentPage.xaml?id="+  
CourseId.ToString(), UriKind.Relative));
```

Parametr *id* je následovně na další stránce v metodě *OnNavigatedTo* získán z tzv. *QueryString*, což je slovník obsahující dvojice předané jako parametr při navigaci.

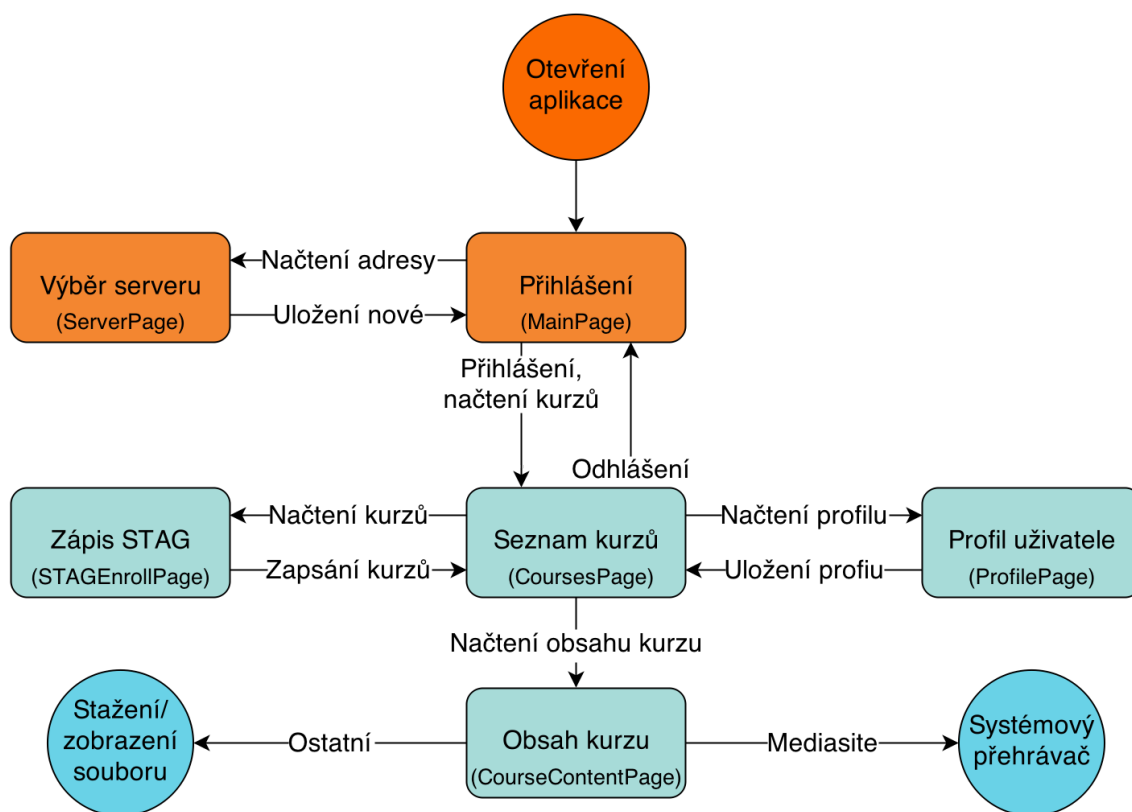
```
string IdString = "";  
if(NavigationContext.QueryString.TryGetValue("id",out IdString))  
    this.CourseId = int.Parse(IdString);
```

3.6 Uspořádání

Aplikace je vytvořena ve frameworku Silverlight pro mobilní telefony. Programy tohoto typu jsou tvořeny jednotlivými stránkami, mezi kterými se uživatel pohybuje. To činí doteky na display a tlačítkem „zpět.“. Jednotlivé stránky aplikace jsou samostatnými třídami, které dědí třídu *PhoneApplicationPage*. Každá třída je rozdělena na dvě části. První částí je xaml soubor, který obsahuje vzhled stránky, uspořádání prvků, jejich hodnoty a popisky. Druhou částí je soubor s příponou „cs“ a ten již obsahuje výkonný kód v jazyce C#.

Jednotlivé stránky dědí množství metod, které jsou vyvolávány asynchronně událostmi. Nejvýznamnější jsou metody vyvolávané příchodem na stránku nebo odchodem na jinou.

- *OnNavigatedTo(NavigationEventArgs e)* – tato metoda je vyvolána při příchodu na stránku. Nehraje roli, zdali uživatel přichází z jiné stránky, nebo se vrací z jiné aplikace. Tato metoda je vhodným místem pro načtení dat z modelu a předání referencí zobrazovací části aplikace.
- *OnNavigatedFrom(NavigationEventArgs e)* – metoda je provedena těsně před odchodem ze stránky. Odchod může být na jinou stránku nebo ven z aplikace. Tato metoda je vhodným místem pro uložení provedených změn a zavření používaných souborů.



Obrázek 3.3: „Mapa“ aplikace

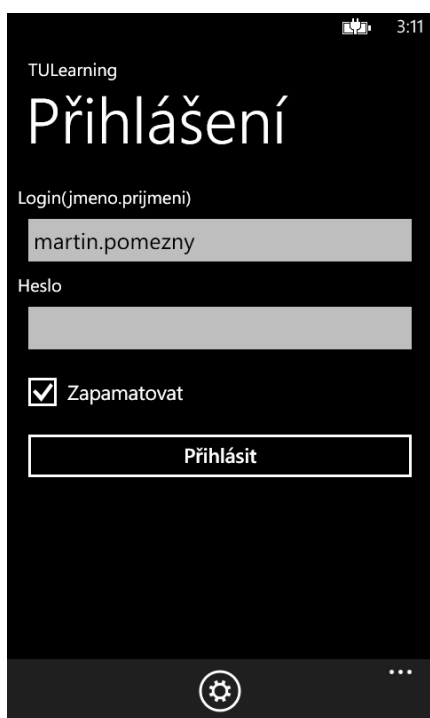
V aplikaci bylo vytvořeno 6 různých stránek. Každá prezentuje uživateli jiná data, ale všechny spolu ladí a snaží se co nejvíce zapadnout do systému, například i tím, že využívají barvu nastavenou jako téma telefonu. Jak jsou jednotlivé stránky propojené, je patrné z obrázku 3.3. V závorkách jsou uvedena jména stránek v aplikaci, šipky popisují co se děje při přechodu z jedné stránky na jinou. Návrat o krok zpět hardwarovým tlačítkem není znázorněn, protože nevyvolává žádné akce ve smyslu činností aplikace (změny se neuloží, ale zahodí). V následujícím seznamu jsou popsány jednotlivé stránky aplikace a jejich činnosti.

- *MainPage* – hlavní stránka aplikace, kterou uživatel vidí hned po spuštění. Obsahuje pole pro zadání jména a hesla k e-learningovému systému (v případě TUL se shoduje s heslem do LIANE). Pod těmito dvěma prvky je také zaškrťovací pole „Zapamatovat“. Pokud jej uživatel před přihlášením zaškrtně, aplikace si uloží jméno a heslo do *Isolated storage* a při příštím spuštění ihned odstartuje proces přihlášení a vyžádání tokenu. V aplikační liště se nachází ikona ozubeného kola, která odkazuje na *ServerPage*. Po kliknutí na tlačítko „Přihlásit“ se zkontroluje, zdali jsou obě povinná pole vyplněna, na což je eventuálně uživatel upozorněn, a provede se přihlášení. Po úspěšném přihlášení se stáhnou základní informace o serveru metodou *GetSiteInfoAsync* třídy *MoodleWebService* a po jejím dokončení stáhne seznam kurzů metodou *GetCoursesAsync* stejné třídy. Přijatá data se uloží do izolovaného úložiště a přesměruje se na *CoursesPage*.
- *ServerPage* – na této stránce může uživatel nastavit adresu serveru, kde běží e-learningový portál Moodle. Pro testování je možné zapsat vlastní adresu. Aplikace obsahuje přednastavené hodnoty pro všechny fakulty TUL a stačí si tedy vybrat, ke kterému e-learningu si uživatel přeje přistupovat. V době tvorby této práce sice fungoval na uvedených adresách Moodle jen na Fakultě mechatroniky a Fakultě pedagogické², dá se tedy předpokládat, že ostatní fakulty postupně budou tento trend přijímat a aplikace tak nebude potře-

²Autor nemá informace o funkčnosti a stavu tohoto portálu

bovat žádné úpravy. Při ukládání hodnoty se regulárním výrazem kontroluje platnost adresy a použitého protokolu(HTTP nebo HTTPS). Aplikace neuloží neplatnou adresu.

- *CoursesPage* – obrazovka obsahuje seznam všech uživatelských kurzů. V hlavě je uvedeno celé jméno přihlášeného uživatele. V seznamu každý prvek sestává ze dvou popisků: prvním je krátké jméno (na TUL ve tvaru ústav/zkratka předmětu), druhým (menším) pak úplné jméno kurzu. Po kliknutí na libovolný kurz aplikace přechází na stránku *CourseContentPage*. Aplikační lišta obsahuje tlačítko pro odhlášení a pro přístup k profilu uživatele. V aplikačním menu se dále nachází odkaz na zapsání předmětů ze STAGu.



Obrázek 3.4: Přihlašovací obrazovka s vyplněným loginem



Obrázek 3.5: Seznam zapsaných kurzů s vysunutým aplikačním menu

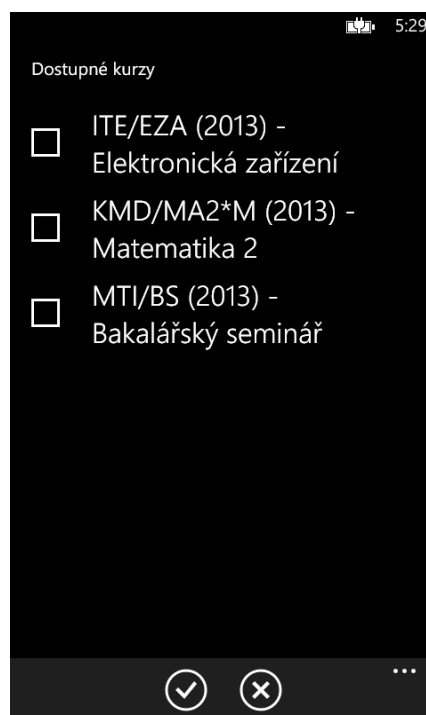
- *STAGEnrollPage* – při vstupu na tuto stránku se nejprve stáhne stránka se zápisem kurzů ze STAGu, následně se pomocí regulárních výrazů najdou kurzy a ty se posléze zobrazí jako zaškrťovací políčka s popisky. Uživatel poté vybere kurzy, o které má zájem. Svou volbu potvrdí stiskem tlačítka „Zapsat“

v aplikační liště. O stavu odeslání dat je informován a následuje návrat zpátky na *CoursesPage*.

- *ProfilePage* – na této stránce vidí uživatel informace o svém profilu a může je rovnou editovat. Needitovatelné údaje, již uvedené v kapitole 2.3.3, jsou zobrazeny jako textová pole s atributem *IsEnabled="false"* a uživateli tak není dovoleno do nich zasahovat. V horní části obrazovky je zobrazena fotografie ze systému Moodle a stahuje se ze serveru jako vše ostatní, asynchronně.



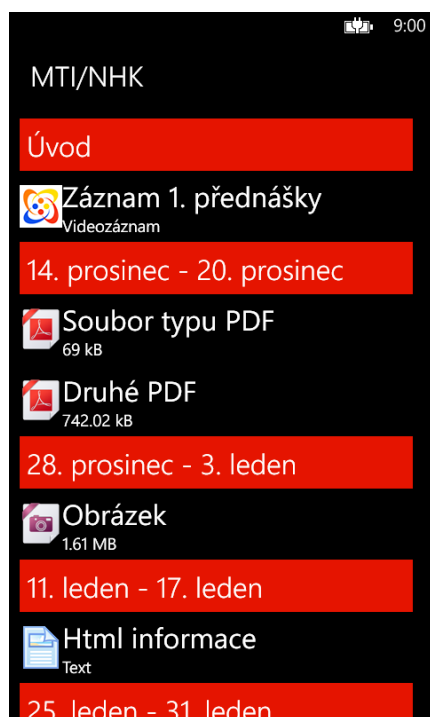
Obrázek 3.6: Profil uživatele



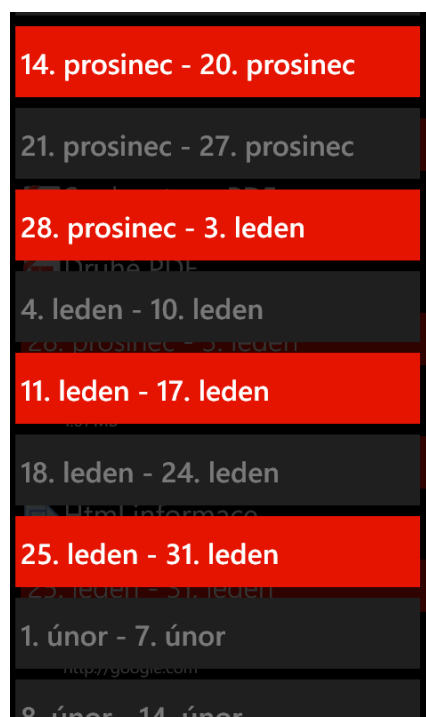
Obrázek 3.7: Zápis kurzů ze STAGu

- *CourseContentPage* – stránka obsahuje v horní části název kurzu, jehož obsah je zobrazován. Jednotlivé sekce kurzu(týdny) jsou graficky rozděleny a jejich název je uveden vždy nad skupinou materiálů. Prázdné skupiny nejsou zobrazeny. Rychlý přechod na konkrétní sekci bez zdlouhavého posouvání obsahu lze provést kliknutím na libovolný nadpis skupiny. Přes celou obrazovku se pak rozbalí přehled všech skupin (viz obrázek 3.9) a po kliknutí na vybranou skupinu aplikace sama přesune pohled nad její obsah. Tento způsob je v systému Windows Phone vžitý a uživatelé jej využívají i například v aplikaci

„Lidé“ integrované v systému. Jednotlivé položky jsou reprezentovány ikonou, která se nestahuje ze serveru, ale vybírá se z ikon uložených v aplikaci podle toho, jakou ikonu by zobrazil systém Moodle. Toto řešení bylo vybráno především proto, že ikony, na která odkazuje webová služba jsou na displayi s vysokým rozlišením příliš malé a při roztažení značně rozmazané. Vlevo od ikony jsou dvě pole textu: horní obsahuje název konkrétního materiálu a je větší. Druhý řádek sestává z informace o souboru, která se určí podle jeho typu. U souborů se ukazuje velikost, u odkazů adresa atd. Po kliknutí se vykoná akce příslušná pro daný typ. Mediasite vyvolává další úkony a zapne přehrávač, jak je uvedeno v kapitole 3.4.2. Otevření, zobrazení a případné uložení ostatních souborů obstará webový prohlížeč integrovaný v systému.



Obrázek 3.8: Obsah kurzu



Obrázek 3.9: Rychlý výběr sekce kurzu

3.7 Distribuce

Distribuce aplikací pro telefony s Windows Phone je zajištěna obchodem s aplikacemi se jménem Windows Phone Store. Ten obsahuje aplikace platformy a uživatelé je mohou komfortně stahovat do svých zařízení a pokud jsou placené, tak za ně rovnou platit. Všechny aplikace v katalogu podléhají certifikaci a jejich kontrolu provádí nejprve automatizované testy a posléze i pracovník na skutečném zařízení.

Instalace aplikace ze souboru přímo do zařízení (tzv. *sideloading*) je umožněna jen vývojářům, kteří telefon odemkli a registrovali na svůj vývojářský účet. Proto by bylo vhodné po všech úpravách a konfiguracích portálu vytvořenou mobilní aplikaci přidat do katalogu. Protože aplikace přináší užitnou hodnotu jen studentům TUL, bylo by vhodné ji nezobrazovat všem uživatelům katalogu, ale volbou „Hide from users browsing or searching the Store“ ji schovat z vyhledávání a přehledu při procházení obchodu. Aplikace bude ale stále v katalogu přístupná, jen je potřeba přesná adresa. Výslednou distribuci může zastat například odkaz na titulní stránce e-learningového portálu nebo QR kód vytištěný na letáku a vyvěšený na nástěnku. Uvedený druh distribuce je běžně využíván u aplikací se specifickou uživatelskou základnou, například návštěvníci festivalů, kulturních akcí, veletrhů, nebo uživatelé služeb jen pro určitý okruh uživatelů.

Závěr

Hlavním cílem práce bylo vytvořit mobilní aplikaci pro systém Windows Phone 8, která umožní studentům pohodlně používat e-learningový portál TUL založený na open-source řešení Moodle. Jeho služeb využívá čím dál tím větší počet studentů a vzhledem ke snaze centralizovat e-learning na celé univerzitě se budou počty ještě zvyšovat. Portál nenabízí v současné době mobilní verzi a je dostupný jen jako webová stránka, která je na mobilním zařízení méně přehledná než na osobním počítači. Spolu s neustále narůstajícím počtem mobilních telefonů nejen mezi studenty se možnost sblížit tyto dvě platformy přímo nabízí.

K funkčnímu nasazení aplikace pro veřejné použití je nezbytně nutné implementovat drobné úpravy a nakonfigurovat portál Moodle pro použití webových služeb. Obojí je podrobně popsáno v této práci. Poté už nebude nic bránit, aby aplikaci začali studenti plně využívat. Spolupráce s Mediasite a streamováním nahraných přednášek do zařízení je plně vyřešena pro stávající stav a není třeba ji nijak konfigurovat nebo upravovat.

Výsledky

Jako výsledek práce vznikla aplikace pro mobilní telefony platformy Windows Phone 8 s názvem *TULearning*. Aplikace je určena studentům TUL, kteří se s její pomocí mohou přihlásit k portálu na svém telefonu a zobrazovat tak informace, procházet předměty, stahovat studijní materiály, prohlížet a editovat svůj profil,

zapisovat předměty které studují ze STAGu a přehrávat záznamy přednášek z portálu Mediasite.

Vzhledem k povaze materiálů a uživatelských dat uložených na serveru byla věnována velká část bezpečnosti, v práci se jí zabývá jedna kapitola. Nad rámec práce bylo částečně zhodnoceno a popsáno i zabezpečení systému Mediasite. K tomu došlo díky implementaci přehrávání videozáznamů přednášek, protože applet poskytující tuto funkci na WP8 nefungoval spolehlivě. Komunikace se serverem je plně šifrovaná a probíhá přes protokol HTTPS. V nastavení lze přepsat adresu a využívat tak i nezabezpečeného protokolu.

Vytvořená aplikace ctí všechna doporučení pro vývoj aplikací na Windows Phone. Uživatelsky je tak intuitivní a snadná na používání. Při volbě vizuálních komponent byl brán ohled na přehlednost a proto jsou použity takové ovládací prvky, aby uživatel netápal, ale na první pohled věděl čeho se má dotknout.

Součástí aplikace je i samostatně použitelná komponenta, která formou třídy jazyka C# implementuje použité funkce webové služby systému Moodle přes protokol REST. Tato komponenta by se mohla za jistých okolností generovat pomocí automatických nástrojů to však nebylo v tomto případě možné a proto byla napsána v rámci práce.

Součástí zadání bylo umožnit práci s aktivitami typu úkol. Tato funkce nebyla nakonec implementována. Hlavní překážkou byla koncepce systému Windows Phone 8, ten je na rozdíl od svého předchůdce (Windows Mobile) zaměřen primárně na konzumaci obsahu a tudíž je na poli práce se soubory značně omezen. Není tedy technicky možné z aplikace přistupovat k souborům na souborovém systému telefonu, jako je tomu možné například na osobních počítačích nebo zařízeních s operačním systémem Android. Aplikace mají oddělené složky do kterých si vzájemně nemohou přistupovat a nic jako úložiště sdílené všemi aplikacemi neexistuje. Výjimku tvoří fotografie, u těch lze aplikaci předat fotografii, kterou uživatel vybral. Vzhledem k tomu, že nahrávání fotografií by na škole technického rázu nebylo využíváno a docházelo

by spíše k matení studentů, kteří by se pokoušeli najít dokumenty nebo soubory, bylo rozhodnuto, že tato funkce by nepřinášela předpokládaný užitek a tak nebyla implementována.

Rozšíření a úpravy

Aplikace byla koncipována jako přístup k e-learningu pokud student nemá k dispozici počítač nebo by použití počítače nebylo vhodné, například tedy při cestování. I proto aplikace nezrcadlí všechny funkce portálu, který je velmi rozsáhlý a díky modulům snadno rozšiřitelný. I proto se dá předpokládat, že časem bude nabízet stále více funkcí. V případě, že by se s postupem času zjistilo, že některá z funkcí studentům chybí, není problém ji do komponenty *MoodleWebService* doprogramovat a rozšířit tak funkčnost aplikace. Ta je totiž vytvořena velmi přehledně a implementaci nových funkcí umožňuje velmi snadno. Ačkoli aplikace obsahuje přednastavené adresy portálů e-learningu všech fakult, drtivá většina jich však stále nefunguje a bylo by tedy vhodné rozšířit myšlenku sjednoceného systému pro e-learning na ostatní fakulty.

Při skutečně hromadném nasazení aplikace by z bezpečnostního hlediska bylo dobré, kdyby token generovaný pro uživatele nebyl stále stejný, ale po nastaveném čase se přepsal novým. To by dále zvýšilo bezpečnost, protože vyzrazení tokenu by potenciálnímu útočníkovi poskytlo jen časově omezenou možnost jej zneužít. Aplikace je k tomuto kroku připravena (token se získává po každém přihlášení).

Pokud by došlo k aktualizaci systému Mediasite a jako prostředek k přehrávání byl aktivován HTML5 přehrávač plně kompatibilní s WP8, bylo by vhodné upravit aplikaci, aby nepřehrávala video jak to činí v současné době, ale aby zobrazila stránku, což by studentům umožnilo zobrazovat i prezentaci k přednášené látce nebo obsah multimediální tabule.

Podobné projekty

Přiblížením systému Moodle mobilním zařízením se již zabývalo vícero projektů, žádný se však nesespecializoval na platformu Windows Phone. Samotný projekt Moodle se pokouší o vývoj aplikace pro mobilní telefony, ale vydal se cestou úpravy HTML a stažitelná aplikace tak akorát obaluje internetový prohlížeč. Při testování se navíc jevila nestabilní a přinášela jen velmi omezené funkce. Implementace nestandardních požadavků TUL (přehrávání Mediasite, zápis ze STAGu) nelze ani od těchto projektů očekávat. Na TUL také vznikla formou diplomové práce aplikace, poskytující podobné funkce pro operační systém Android.

Reference

- [1] PETZOLD, Charles. *Programming Windows Phone 7*. Redmond, WA: Microsoft Press, 2010, p. cm. ISBN 978-073-5643-352.
- [2] Rosoff, Matt. *For The First Time Ever, Smartphones Outsold PCs Last Quarter* [online]. 3. 2. 2012. URL:<<http://www.businessinsider.com/for-the-first-time-ever-smartphones-are-outselling-pcs-2012-2>>
- [3] Columbus, Louis. *2013 Roundup of Smartphone and Tablet Forecasts & Market Estimates* [online]. 17. 1. 2013.
URL:<<http://www.forbes.com/sites/louiscolumbus/2013/01/17/2013-roundup-of-mobility-forecasts-and-market-estimates/>>
- [4] Moodle, komunita. *Moodle statistics* [online]. 2013.
URL:<<https://moodle.org/stats>>
- [5] Moodle, komunita. *SOAP JAVA server could generate a JAX-WS compatible WSDL* [online]. 18. 8. 2011.
URL:<<https://tracker.moodle.org/browse/MDL-28989>>
- [6] Ewald, Tim. *The Argument Against SOAP Encoding* [online]. 10/2002.
URL:<<http://msdn.microsoft.com/en-us/library/ms995710.aspx>>
- [7] Moodle, komunita. *Service add admin page: should be able to enter the unique shortname of a service* [online]. 17. 10. 2011.
URL:<<https://tracker.moodle.org/browse/MDL-29807>>
- [8] Moodle, komunita. *WS core_user_update_users should allow to update own profile* [online]. 28. 4. 2013.
URL:<<https://tracker.moodle.org/browse/CONTRIB-4282>>
- [9] Newton-King, James. *Json.NET 5.0 Release 4 – Performance* [online]. 25. 4. 2013. URL:<<http://james.newtonking.com/archive/2013/04/25/json-net-5-0-release-4-performance>>
- [10] Open Source Initiative. *The MIT License (MIT)* [online]. 2007.
URL:<<http://opensource.org/licenses/MIT>>